Real-time landscape-size convective clouds simulation and rendering

Prashant Goswami, Fabrice Neyret

Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2017)
F. Jaillet and F. Zara (Editors)

Real-time landscape-size convective clouds simulation and rendering

Prashant Goswami[†1] and Fabrice Neyret[2‡]

[1]BTH Sweden
[2]INRIA, Univ. Grenoble Alpes/LJK, CNRS/LJK

Abstract

This paper presents an efficient, physics-based procedural model for the real-time animation and visualization of cumulus clouds at landscape size. We couple a coarse Lagrangian model of air parcels *with a procedural amplification using volumetric noise. Our Lagrangian model draws an aerology* i.e., *the atmospheric physics of hydrostatic atmosphere with thermodynamics transforms, augmented by a model of mixing between parcels and environment. In addition to the particle-particle interactions, we introduce particle-implicit environment interactions. In contrast to the usual fluid simulation, we thus do not need to sample the transparent environment, a key property for real-time efficiency and scalability to large domains. Inheriting from the high-level physics of aerology, we also validate our simulation by comparing it to predictive diagrams, and we show how the user can easily control key aspects of the result such as the cloud base and top altitude. Our model is thus fast, physical and controllable.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: clouds, natural phenomena, real-time, animation, amplification, GPU

1. Introduction

Clouds are ubiquitous in outdoor naturalistic scenes, especially cumulus clouds which make interesting and conspicuous patterns. Thus they form an important component of many outdoor graphics applications, such as video games and flight simulators. Since these applications are becoming increasingly realistic, covering large and distributed landscapes, a similar trend is expected from their clouds which is currently not the case. Moreover, clouds could be one of the secondary features in these applications. Therefore, their animation and rendering should take only a small fraction of the total frame time.

The requirements are contradictory. Purely procedural models like [Ebe97, STW11, MMPZ12] cannot convey the animated look of cumulus clouds which is a complex and evolving volumetric phenomena. On the other hand, traditional fluid simulation methods like computational fluid dynamics (CFD) simulations are not affordable in real-time, especially at landscape scale or with thin 3D details.

The evolving cloud simulation through huge Eulerian solvers is challenging even for physicists, due to the fact that small-scale and large ranges are required at the same time causing the numerical blending to affect the thermodynamics. In the scope of real-time graphics, cloud simulations do exist, for example based on Eulerian physics [HBSL03]. However, they address only very limited volumes or show very coarse resolutions. One of the big disadvantages both with CFD and Eulerian methods is that most of the sky volume is filled with invisible air that needs to be stored and computed. Cloud simulation has also been proposed using Lagrangian framework [FBDY15] but such methods are dependent on a high particle count to achieve realistic resolution.

We propose a hybrid *physics-inspired procedural* method for the real-time simulation and rendering of detailed clouds. The main contributions of our approach are:

- We introduce *smoothed-particle hydrodynamics* (SPH) based Lagrangian framework for the cloud physics, operating at the macro-level on large particles called *thermals* making it computationally inexpensive.
- To avoid sampling the ubiquitous clear sky around clouds, we extend the Lagrangian formalism to account for the hybrid particle-implicit environment which is the key for real-time and scalability. For closure, we also introduce a new formulation of cloud-environment mixing (entrainment/detrainment).
- The use of Lagrangian formalism, aerology physics and separation of environment from the dynamics brings more robustness, offers meaningful high-level handles for the user control on environment profile and evolution, cloud base and top.
- For visualization, we treat these thermals as procedural units containing hypertexture. The volumetric texture inside a ther-

† prashant.goswami@bth.se
‡ fabrice.neyret@inria.fr

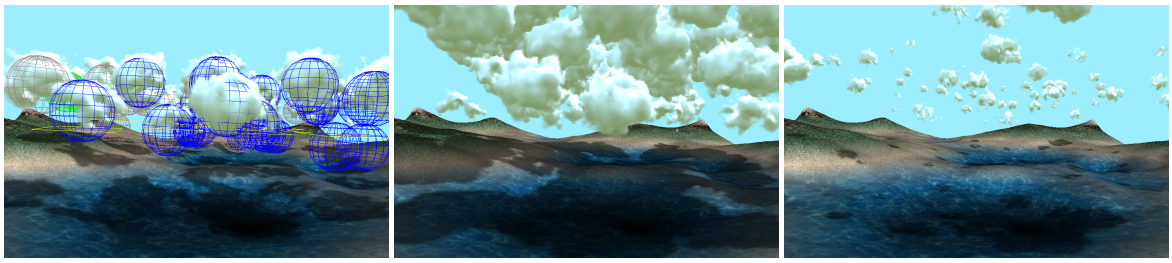


Figure 1: *Cloud life cycle demonstrated using 75 parcels resulting from the space-time distributed emission from the ground with probabilities depending on the local wetness and temperature. Clouds condense when reaching the condensation level altitude, rise up to equilibrium altitude, then decay and die due to the effects of entrainment and detrainment.*

mal operates at the micro-level and is produced based on its macro physical state.

The model lightness makes possible the first real-time, landscape-size convective cloud interactive simulation, with rendering capturing small-scale details. It is at the same time physically validated and controllable, which is important for applications aiming at the contradictory requirements of natural look, predictable aspect and performance. This paper is the detailed exposition of our earlier published poster [GN15].

2. Related Work

Much work exists on fluid simulation in the graphics literature, most of which is out of reach of real-time constraints for cloud animation or rendering. Moreover, these methods often account for a very limited volume since transparent regions generally cost as much as the visible ones. There do exist a few multiscale simulation schemes which make use of multigrid and finite elements but the performance is still far from real-time. Moreover, cloud physics includes thermodynamics which is a crucial ingredient (and constraint) since advected scalars are reactive; clouds are not just smoke.

The existing approaches for cloud animation in computer graphics follow different strategies. In [DKY*00] the cloud evolution is simulated using cellular automaton where the dynamics are specified using simple transition rules. [DYN06] proposes a method for animating clouds surrounding the earth at planetary level based on high and low pressure regions. Mostly [KVH84, Ney97, HBSL03] account for thermodynamics and are thus directly relevant to us. However, they come with all the problems of Eulerian simulations like limited domain, coarse resolution, harsh initial conditions, energy damping, numerical dissipation, lack of control, etc. Eulerian approaches also suffer a lot from low resolution since artificial blending modifies the physics, damping energy, etc. misrepresenting important characteristics such as vorticity and free boundaries.

Lagrangian approaches like SPH [MCG03] are considered more reliable on moving features, even at low resolutions. [FBDY15] recently proposed a particle-based method for cloud simulation based on position-based fluids [MM13]. Their approach is based on physics and thermodynamics and relies on sampling air in a limited domain which involves merging and splitting SPH particles in transparent and cloudy regions respectively. Consequently, a high particle count is necessitated wherein physics computation takes up the bulk of computational time. and visualization is achieved as a secondary, offline step.

A key issue with sampling the transparent air is that equations are only a part of the problem; initial conditions, state of the whole atmosphere and evolving boundary conditions like humidity and temperature on the floor are equally crucial. It is therefore, advisable to separate inputs comprising the tropospheric environment (for which the hydrostatic hypothesis is very reasonable), from simulated outputs *the clouds*.

In terms of the control of fluids, the existing works aim at artistic control of fluid animation or pure proceduralism [SSEH03, DKNY08, WBC08]. This is totally different to controlling a few high-level, visible characteristics emerging from the simulation such as nebulosity, size and altitude range of clouds. Contrary to these methods, we rely on the high-level physics to simply govern these emerging characteristics. Some work on real-time visualization of clouds is also based on 2D textural cloud layer or sprite-based clouds. The rendering part in our technique could benefit by using one or more of the existing methods [ES00, BNM*08, Gar85]. The reader is referred to [Ye14] for a more detailed insight into the rendering methods for clouds.

3. Atmospheric Physics

Rising air parcels are subjected to the laws of thermodynamics, buoyancy and mixing with the surrounding environment, which we briefly describe here.

Thermodynamics: Humidity is characterized by the *mixing ratio* ω which is the weight of vapor in grams per kilogram of dry air. The *saturated mixing ratio* ω_s is the limit beyond which the extra vapor condenses into liquid water, releasing *latent heat* $L(= 2260kJ/kg)$ per gram of condensed mass. The temperature change in a parcel with mass m due to condensation (or vaporization) of Δm of vapor is

$$\Delta T = \alpha \frac{L}{C_p} \frac{\Delta m}{m} \qquad [Kelvin] \tag{1}$$

considering adiabatic conditions. Here $C_p = 1.006$ kJ/kg.K is the specific heat of air and α is a constant (similar to [FBDY15]) which is empirically set to 0.3.

The saturated mixing ratio is obtained from the pressure P (which changes with altitude h) and the *saturated vapor pressure*

e_s (which depends only on the temperature T in Kelvin):

$$P = P_0 e^{-0.143h} \quad [hPascal] \tag{2}$$

$$e_s = 6.11 \times 10^{\frac{7.5(T-273.15)}{T-35.45}} \quad [hPascal]$$

$$\omega_s = 621.97 \frac{e_s}{P-e_s} \quad [\frac{g}{kg}] \tag{3}$$

where $P_0 \approx 1013.25$ hPascal.

Temperature, buoyancy and stability: Two key ingredients here are the atmospheric profile (temperature and humidity with altitude) and the initial conditions on the ground where the thermal bubbles start from. With the help of these parameters, one can obtain the altitude of cloud base known as the condensation level where the saturation starts ($\omega = \omega_s$) and the top where the buoyancy vanishes and the parcel reaches a stable equilibrium. To ease the comparisons and computations, atmospheric physicists introduced several alternate notions of temperature. While *potential temperature* $\Theta = T\left(\frac{P_0}{P}\right)^{0.286}$ and *virtual temperature* $T_v = T(1+0.61\omega)$ account for the change of pressure and moisture respectively on parcel, *virtual potential temperature* Θ_v accounts for the combined effect of both.

$$\Theta_v = \Theta(1+0.61\omega) \quad [Kelvin] \tag{4}$$

With these definitions, Archimedes' buoyancy force per unit mass can be simplified to

$$\mathbf{f}_{buoy} \approx (\frac{\Theta_v^{parcel} - \Theta_v^{atm}}{\Theta_v^{atm}})g \quad [\frac{Newton}{kg}] \tag{5}$$

(by using ideal gas law $P \propto \rho T$).

Parcel-environment interactions: Mixing has a profound effect on the parcels since it modifies water and air content as well as the temperature. In this context, *entrainment* is engulfing relatively dry environment air in the cloudy parcel while *detrainment* is leaking of cloud mass into the environment. A commonly accepted formulation often assumes the form $\frac{\delta m}{m \delta z} = \varepsilon - \delta$ where δm is the change in the mass of parcel, z its altitude and ε and δ are entrainment and detrainment constants respectively [SMST04]. Parcels are also subjected to friction or drag force $\mathbf{f}_{airDrag} = -\frac{1}{2} C_D \rho v^2 A$ where v is the relative velocity of parcel, A its reference surface and C_D is the drag coefficient.

From Laplace's law $PV^\gamma = constant$ ($\gamma \approx 1.4$ for air) and accounting for possible mass flux and pressure change, the parcel's volume is estimated using

$$V_2 = V_1 (\frac{P_1}{P_2})^{\frac{1}{\gamma}} (\frac{m_2}{m_1}) \quad [m^3] \tag{6}$$

SPH: Since our method is based on the discretization of fluid physics with particles, we need to introduce the SPH formalism. The particles influence each others' movements through pressure and viscosity forces. In SPH a weighting kernel W within a given support radius sr is associated to the particles and pressure is solved through a state law around equilibrium. Density is computed using the neighborhood set over support radius (summation over j) for each particle

$$\rho_i = \sum_j m_j W(\mathbf{x}_i - \mathbf{x}_j, sr) \tag{7}$$

where m_i refers to the mass of i^{th} particle and $\mathbf{x}_i$ its position. Pressure follows the spring mass formulation, where fluid rest density is given by ρ_0 and K is the stiffness constant.

$$p_i = K(\rho_i - \rho_0) \tag{8}$$

The pressure force is derived from the pressure using Navier-Stokes equation

$$\mathbf{f}_{pressure_i} = -\sum_j m_i m_j (\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}) \nabla W(\mathbf{x}_i - \mathbf{x}_j, sr) \tag{9}$$

and the viscosity force (or the drag between particles) using relative velocity between neighboring particles (μ is viscosity coefficient)

$$\mathbf{f}_{viscosity_i} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, sr) \tag{10}$$

We refer the reader to [MON05] for the weighting kernels employed in our method and for more in-depth survey of SPH.

4. Model Overview

We adopt a two-scale model, with large particles called *parcels* acting as the physics units and simultaneously containing volumetric texture. Through this decoupling between the macro and micro levels, a physics-inspired procedural method is made possible. We introduce these levels in the following and cover them in detail in section 5 and section 6 respectively.

1. *Macro level* : On the physics front, our particle solver inherits most of the SPH fluid simulation formalism with the difference that our parcels are large and correspond to the thermals in [SSEH03], with very little overlapping. Unlike existing methods, the environment profile is modelled implicitly and consists of piecewise linear potential temperature and humidity. This enables us to have particles only for thermals and clouds and not for the whole sky, thereby avoiding expensive physics and value storage for transparent regions. For closure, we introduce a model of particle-implicit environment interaction, hybridised with SPH.
2. *Micro level* : On the visualization front, our amplification relies upon on-the-fly procedural hypertexture generation within the parcels and a GPU volume ray-tracer. We evaluate a per-parcel *filling ratio* factor used to clamp a portion of the volume to account for the internal heterogeneity occurring based on the amount of condensed water content. This is inspired by the fact that condensation first occurs on the top of the parcel and at cloud decay lower average droplet density corresponds to a fractal rather than a continuous haze.

5. Macro scale

We rely on the classical Lagrangian model for the particle-particle interaction. The inter-parcel interaction is captured by applying repulsive pressure force $\mathbf{f}_{pressure}$ (Equation 9) that prevents parcels from getting too close to each other and the viscosity force $\mathbf{f}_{viscous}$ (Equation 10) that accounts for the drag between the parcels. Besides the SPH forces (in our model), cloud parcels are also subjected to the thermodynamic rules. Thus in the following, we describe the interaction of parcels with the implicit environment, inter-parcel mixing and the buoyant forces acting on parcels, thereby completing the total parcel physics leading to cloud formation.

Algorithm 1 Parcel physics computation

while (animating) **do**
  update neighborhood grid
  for all parcels p **do**
    – get neighbors(p)
    – compute density(p), pressure(p) (7,8)
    – compute pressure force $\mathbf{f}_{pressure}$, viscosity force $\mathbf{f}_{viscosity}$ (9,10)
    – compute ambient atmospheric pressure force $\mathbf{f}_{atmPres}$ (2)
    – compute entrained mass Δm_e and update potential temperature Θ' (13,14)
    – eject detrained mass Δm_d (15)
    – compute water mass exchange between neighboring parcels Δm_p
    – update volume V (6)
    – update mixing ratio (liquid/vapor content) and potential temperature change $\Delta\Theta$ (3,1)
    – compute buoyant force $\mathbf{f}_{buoy}$ (5)
    – apply $\mathbf{f}_{parcel}$ and wind to move the parcel (17)

5.1. Interactions with the implicit environment

In our model, particles cover only a part of the atmosphere which still must represent a continuum of all physical interactions. We also therefore need to estimate the contribution of the implicit (*i.e.,* unsampled) environment on the area of parcels exposed to the atmosphere. Thus to these particle-particle interactions, we have to add forces corresponding to the equivalent terms in particle-environment interactions, $\mathbf{f}_{atmPres}$ and $\mathbf{f}_{airDrag}$ respectively.

Atmospheric force $\mathbf{f}_{atmPres}$ is applied to the exposed surface area of each parcel which is in direct contact with the atmosphere. The exposed area (denoted by $\mathbb{S}$) is determined by coarsely tessellating each parcel into quadrilaterals and determining the set of quadrilaterals that do not lie inside any other parcel. Instead of querying all the parcels, a quadrilateral needs to perform this test only on the SPH neighbors of the parcel it belongs to. Atmospheric pressure is obtained using Equation 2 and is applied to each tessellated quadrilateral along its normal weighted by the area. $\mathbf{f}_{atmPres}$ needs an additional weighting factor, which we obtain through calibration based on our experiments.

5.2. Mixing with the environment

The parcels interact with the surrounding environment through the process of *entrainment* and *detrainment*. The total amount of entrained air in a parcel during time interval Δt is a function of both its exposed area $\mathbb{S}$ and the velocity, which in essence gives the swept volume of the parcel through the external relatively dry air. Even if a parcel is globally motionless, small-scale turbulence pursues the mixing; otherwise clouds would not decay. To account for this, we thus add to the macroscopic drag a small turbulence-related term $\delta\mathbf{v}_t$. Furthermore, fractal nature of clouds makes the real cloud-air contact surface larger than one for the macroscopic sampling primitives. For this we assume proportionality coefficients, K_e for entrainment and K_d for detrainment, which is the surface multiplier to account for subgrid fractal surface.

Entrainment entails the mixing of external air Δm_e^a (Equation 11), which also carries with it the water vapor Δm_e^w (Equation 12) already present in it (computed using mixing ratio), with the parcel. The total mass gain Δm_e is obtained by summing these two quantities (Equation 13). To incorporate it in our formulation, we modify the standard entrainment and detrainment constants to $\varepsilon' = \varepsilon\, r$ and $\delta' = \delta\, r$, where r is the radius of the parcel.

$$\Delta m_e^a = K_e \varepsilon' A \rho^{atm} \|\mathbf{v} + \delta\mathbf{v}_t\| \Delta t \quad (11)$$

$$\Delta m_e^w = \omega_s \Delta m_e^a \quad (12)$$

$$\Delta m_e = \Delta m_e^a + \Delta m_e^w \quad (13)$$

Here ε is the entrainment rate taken from [SMST04] and A is the exposed surface area obtained from the computed $\mathbb{S}$. The entrained mass comprising both dry air and vapor is at a different temperature than the parcel. The equilibrium temperature Θ' is computed by mixing the influx with the existing parcel contents m, see Equation 14.

$$\Theta' \approx \frac{m\Theta^{parcel} + \Delta m_e \Theta^{atm}}{m + \Delta m_e} \quad (14)$$

Subsequently, the quantity of detrained content Δm_d (Equation 15) is determined using a similar expression as for entrainment albeit with a few changes. ρ^{atm} in Equation 11 is substituted with the average density ρ in the parcel to compute Δm_d^a. Assuming a uniform mixing of vapor in the parcel, an equal proportion of water mass Δm_d^w is ejected together with Δm_d^a. Furthermore, ε is also replaced by δ, the detrainment constant. The total mass detrained from a parcel is given by Equation 15.

$$\Delta m_d = \Delta m_d^a + \Delta m_d^w \quad (15)$$

5.3. Inter-parcel mixing

Much as the parcels exchange mass with the external environment through their exposed surface, a similar process takes between the neighboring parcels through the contact surface. The inter-parcel mass exchange Δm_p is obtained using a simple diffusion process from high concentration parcel to lower ones surrounding it.

With this, the combined mixing occurring to a parcel could be summed up as:

$$\Delta m = \Delta m_e - \Delta m_d + \Delta m_p \quad (16)$$

5.4. Buoyant force

In addition to aforementioned forces, buoyant force $\mathbf{f}_{buoy}$ is added to each parcel which is a key component in parcel rising and cloud formation. With the computation of updated potential temperature Θ' (Equation 14) inside the parcel, buoyant force follows from Equation 5 based on the temperature difference with the surrounding environment.

With this, the total force acting on a parcel is:

$$\mathbf{f}_{parcel} = \mathbf{f}_{pressure} + \mathbf{f}_{viscous} + \mathbf{f}_{atmPres} + \mathbf{f}_{airDrag} + \mathbf{f}_{buoy} \quad (17)$$

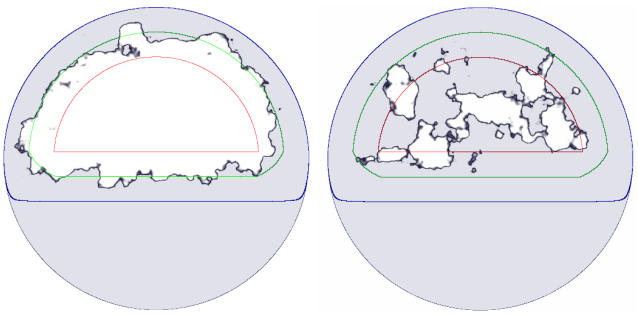


Figure 2: *Left: hypertexture in the shell (between the blue and the red curves) of sphere ∪ half-space above condensation level, right: with filling ratio f=50%.*

6. Micro Scale

The micro level deals with the procedural generation of cloud details on top of the existing macro structure of parcels. The hypertexture is computed on-the-fly, and is controlled by a few macro-level handles such as particle radius, condensation level plane and filling ratio which are themselves derived from the ongoing parcel physics. This is detailed in subsection 6.1. We employ GPU-based ray-tracing to efficiently render our clouds as explained in subsection 6.2.

6.1. Volumetric cloud texture

From the local list of parcels intersecting a given point in space, we employ the hypertexture framework of [PH89] to compute the union of clamped hypertextured parcels (see Figure 2):

- The pseudo-distance is computed as $\cup_i(sphere_i \cap halfSpace_i)$ where the sphere corresponds to the parcel and the half-space corresponds to clamping by the condensation level point plane. Since our SPH kernels have low super-imposition, we use spheres larger than these kernels for nice looking unions. Note that we can stop the $\cup_i$ loop as soon as it saturates to 1.
- The sphere and half-space shell thickness is a user-tunable constant, but is modulated by the parcel filling ratio f to account for the cloud decay so that the noise texture gets sponge-like and erodes with the average density proportional to f. The density is then computed based on the noise and distance function (where k is a constant):

$$\rho(P) = smoothstep(f.(dist(P)+1)-1 + k.noise(P)) \quad (18)$$

The factor f estimates the percentage of a parcel filled with condensed (opaque) material. The idea is that the filling is not homogeneous. Condensation starts at condensation level (denoted by half-space) and grows making the parcel more opaque while decaying carves a sponge-like texture. Therefore, in practice an average droplet density corresponds to a mix of clear and opaque fractions. This relative volume of opaque fraction is the filling ratio. We estimate it as $f = \frac{\omega_{cloud}-\omega_s}{V^{air}_{1kg}.q_{l0}}$ where V^{air}_{1kg} is the volume of 1kg of air at parcel altitude and q_{l0} is the droplet density in the opaque fraction.

6.2. Ray-tracing clouds

The real-time volumetric details generated from subsection 6.1 are visualized using volume ray casting. Our simple physics engine can easily be coupled to a reasonably minimal volume raycasting fragment shader. To this end, we have used the *uniform buffer object (UBO)* which allows us to pass arrays to the fragment shader. We transfer the entire grid together with the list of condensed parcels and the related attributes (parcel size, filling ratio, plane position) as *UBO* to the GPU.

7. Results and discussion

The proposed method was implemented using C++, OpenGL version 4.0 and tested on an Intel Core 3.4 GHz machine with 16 GB RAM and an NVidia GeForce GTX 970 graphics card on a display window of 730×545 pixels.

The number of parcels is user chosen and can vary from single to many. Our model allows realistic, landscape-scale cloud effects by using just few macro primitives. The physics part therefore, is a lightweight computation which can be carried out on the CPU itself. However, this is not a limitation of our method as in case where larger number of macro primitives are desired, physics computation could be easily and efficiently transferred to GPU as laid out in [GSSP10]. We also cast simplified shadow of clouds on the OpenGL terrain. This is achieved by shooting rays from the light source towards the ground to collect cloud opacities in a 2D texture in the first pass and blending them with the scene at high resolution in the second pass.

The parcel tessellation for atmospheric pressure and mixing computation is kept coarse (18×9) in our case which provides a good balance between accuracy and computational effort. We set the environment dimensions to $10km \times 10km \times 10km$ where the simulation and visualization is carried out, although it is flexible and user modifiable (and could as well be irregular). In terms of Lagrangian parameters, density of air is taken to be around $\rho_0 = 1.28kg/m^3$, stiffness constant $K = 100$ and a low viscosity coefficient $\mu \approx 0.01$ is chosen.

7.1. Aerology validation

Single-parcel kinematics is verified against aerology diagram for the following important parameters:

- *Lapse rates*: As computed directly from Figure 4a, the *dry adiabatic lapse rate* comes out to be around -10 C/km (slope between points A and B), whereas the *moist adiabatic lapse rate* is about -5 C/km (average slope between points B and D). These values are close to the actual figures, as given in [Mih11]. Air parcels that get saturated as they rise cool at a smaller rate than the dry adiabatic lapse rate due to the heat released by the condensation of water vapor (B onwards). The wet adiabatic lapse rate varies with the altitude which is why we take into account the average value.
- *Equilibrium altitude*: In the case of no mixing with the atmosphere, the parcel reaches to a maximum altitude and comes to an equilibrium height after a series of damping oscillations, as is visible in Figure 4a, see also Figure 3a. The maximum altitude lies close to the point where parcel temperature intersects the atmospheric temperature profile. The parcel continues moving further up till its velocity is reversed by the negative buoyancy

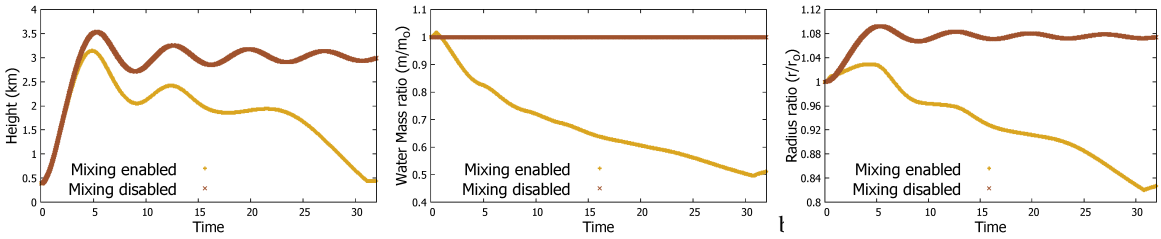
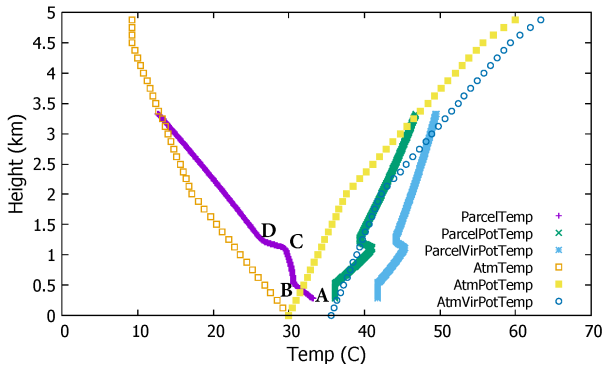
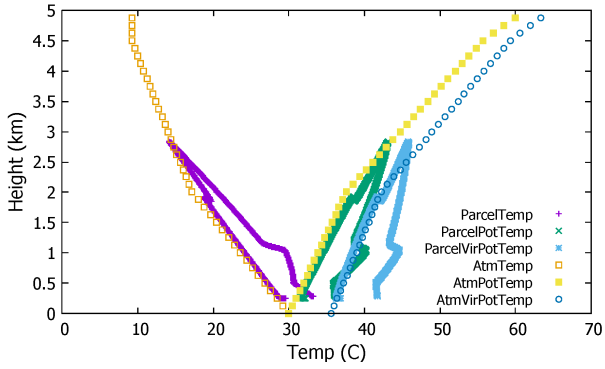


Figure 3: *Attributes of a parcel varying with time in our simulation, with mixing enabled and disabled: (left) parcel position, (middle) parcel water mass ratio relative to its initial value, (right) parcel radius ratio relative to its initial value.*



(a) *Mixing disabled*



(b) *Mixing enabled*

Figure 4: *Parcel vs. environment temperatures (T, Θ, Θ_v) during single parcel simulation with (a) mixing disabled (b) mixing enabled.*

force. However, mixing makes a parcel lose its water content, thereby changing the internal dynamics. During this decaying process the parcel undergoes oscillatory motion too as it alternates between rising and sinking buoyant force [Hob93], which is also validated by our simulation as shown in Figure 4b and Figure 3a. Note that while the parcel follows the same temperature trajectory during its oscillations in absence of mixing (Figure 4a), the temperature profile is altered with each oscillation if the mixing is enabled (Figure 4b).

- *Mass and radius*: The variation of parcel water mass and radius relative to its initial values is as demonstrated in Figure 3b and Figure 3c respectively. The initial mass gain with mixing enabled is explained by the fact that as per the model from [SMST04], detrainment begins at condensation level while entrainment occurs throughout. It is for the same reason that the parcel begins to gain mass towards the end of its lifetime as it reaches the condensation level again. Eventually, the parcel exhausts its water content significantly by leaking it to the atmosphere. The parcel radius grows as it ascends due to the decrease in atmospheric pressure. However, in the case of mixing with atmosphere the parcel radius shrinks due to the combined effect of water loss and descent.

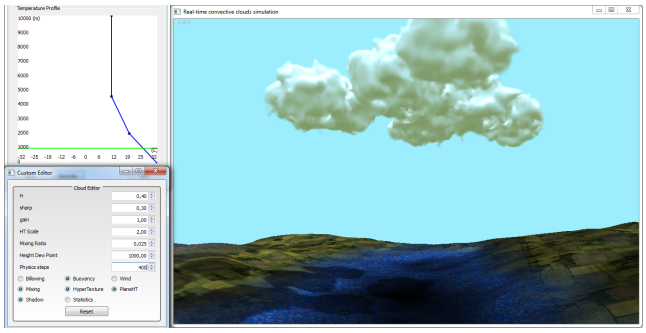


Figure 5: *Our interactive tool allows the user to experiment with intuitive parameters to generate plausible evolving fields of cumulus clouds.*

7.2. User control

The condensation level altitude and the nature of clouds obtained from parcels is dependent on the initial mixing ratio and the temperature profile. The cloud top is governed by the *temperature inversion*, the altitude at which the temperature begins to increase and hence the clouds parcel cease to rise further up. Though the default implicit environment is predefined in our system, our tool (Figure 5) gives the user an easy and intuitive handle to:

- alter the temperature profile graphically
- select condensation level
- choose mixing ratio of the atmosphere

The system simulates the physics with the user requirements and the clouds are generated if the state is physically plausible.

7.3. Cloud life cycle

Our parcel generator automatically generates parcels with a frequency proportional to the heat and humidity content of the region.

# parcels	with FDM normals			without FDM normals		
	physics	render	fps	physics	render	fps
8	6%	89%	128	4%	78%	235
14	6%	89%	77	6%	91%	160
34	8%	87%	24	8%	89%	63
85	9%	82%	7	9%	88%	21

Table 1: *Performance split-up between the physics part on the CPU and the rendering part on the GPU together with the frame rates per second (fps).*

Several time-consistent plumes of parcels could be emitted one after the other piling up on top of each other and giving rise to interesting cloud patterns. Figure 1 illustrates the complete life cycle of clouds as parcels from their birth to death. Clouds often die out in order of the sequence of their creation and also depending on their exposure to the outside dry air.

7.4. Performance analysis

We have tried up to 85 dynamically produced parcels where the simulation together with the rendering continues to be interactive. The performance analysis of our system is summarised in Table 1, where the percentages are given relative to the total time spent in the system per frame. In there we also compare our performance with and without computation of normals (for voxel shading) using finite difference method (FDM) for rendering. Our experiments suggest that rendering without finite differences normal is significantly faster than with normals. The only expensive part in parcel physics is the computation of area exposed to implicit environment on the tessellated sphere. On the other hand, hypertexture production and raycasting based visualization takes a larger share of overall computational time as compared to physics.

In Figure 6, a realistic and distributed cloud cover is obtained with just 14 parcels. Further, interesting cloud patterns are obtained by taking the union of density parcels in the overlapping regions as shown in Figure 7. In addition to the above mentioned experiments, being a Lagrangian-based technique our method can easily deal with the effects of wind, parcel-obstacle collision which could influence the cloud movement and formation. Furthermore, the atmospheric temperature and humidity profile can be varied as a function of time of the day to influence cloud formation. These aspects of our method are demonstrated in the accompanying video.

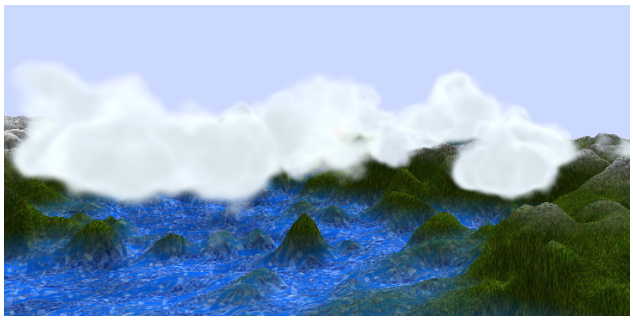


Figure 6: *Parcel physics and hypertexture generated with just 14 parcels leading to fast and yet, realistic visualization.*

Figure 7: *Interesting patterns resembling cumulus clouds are obtained by taking union of the densities in the overlapping parcel regions.*

8. Conclusions and future work

We have presented an efficient, physics-based procedural hybrid method for real-time landscape-size simulation and visualization of cumulus clouds. The use of Lagrangian framework at the macro parcel level makes the physics computationally inexpensive. The introduction of the implicit atmosphere eliminates the need to sample transparent air and accounts for the parcel-air interaction and mass exchange. The real-time visualization of clouds is achieved by raytracing the procedural amplification of volumetric noise on the GPU. Our model is controllable by the user and predictive, and can be validated against the areology diagram for basic parameters.

For future work, we would like to add more complex physics in order to permit billowing of cumulus clouds, and study hierarchical Lagrangian simulation for even larger scale scalability.

References

[BNM*08] BOUTHORS A., NEYRET F., MAX N., BRUNETON E., CRASSIN C.: Interactive multiple anisotropic scattering in clouds. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2008), I3D '08, ACM, pp. 173–182. URL: http://doi.acm.org/10.1145/1342250.1342277, doi:10.1145/1342250.1342277. 2

[DKNY08] DOBASHI Y., KUSUMOTO K., NISHITA T., YAMAMOTO T.: Feedback control of cumuliform cloud formation based on computational fluid dynamics. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 27*, 3 (2008). URL: http://nis-ei.eng.hokudai.ac.jp/~doba/projects/cloud_control/cloud_control.htm. 2

[DKY*00] DOBASHI Y., KANEDA K., YAMASHITA H., OKITA T., NISHITA T.: A simple, efficient method for realistic animation of clouds. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH, ACM Press/Addison-Wesley Publishing Co., pp. 19–28. URL: http://dx.doi.org/10.1145/344779.344795, doi:10.1145/344779.344795. 2

[DYN06] DOBASHI Y., YAMAMOTO T., NISHITA T.: A controllable method for animation of earth-scale clouds. In *Proc. of CASA* (2006), pp. 43–52. URL: http://nis-ei.eng.hokudai.ac.jp/~doba/papers/casa06.pdf. 2

[Ebe97] EBERT D. S.: Volumetric modeling with implicit functions: A cloud is born. In *ACM SIGGRAPH 97 Visual Proceedings: The Art and Interdisciplinary Programs of SIGGRAPH* (New York, NY, USA, 1997), SIGGRAPH, ACM, pp. 147–. URL: http://doi.acm.org/10.1145/259081.259233, doi:10.1145/259081.259233. 1

[ES00] ELINAS P., STÜRZLINGER W.: Real-time rendering of 3D clouds. *J. Graph. Tools 5*, 4 (Oct. 2000), 33–45. URL: http://dx.doi.org/10.1080/10867651.2000.10487531, doi:10.1080/10867651.2000.10487531. 2

[FBDY15] FERREIRA BARBOSA C. W., DOBASHI Y., YAMAMOTO T.: Adaptive cloud simulation using position based fluids. *Computer Animation and Virtual Worlds 26*, 3-4 (2015), 367–375. URL: http://dx.doi.org/10.1002/cav.1657, doi:10.1002/cav.1657. 1, 2

[Gar85] GARDNER G. Y.: Visual simulation of clouds. *SIGGRAPH Comput. Graph. 19*, 3 (July 1985), 297–304. URL: http://doi.acm.org/10.1145/325165.325248, doi:10.1145/325165.325248. 2

[GN15] GOSWAMI P., NEYRET F.: Real-time landscape-size convective clouds simulation. In *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2015), I3D, ACM, pp. 135–135. URL: http://doi.acm.org/10.1145/2699276.2721396, doi:10.1145/2699276.2721396. 2

[GSSP10] GOSWAMI P., SCHLEGEL P., SOLENTHALER B., PAJAROLA R.: Interactive SPH simulation and rendering on the GPU. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2010), SCA, pp. 55–64. URL: http://dl.acm.org/citation.cfm?id=1921427.1921437. 5

[HBSL03] HARRIS M. J., BAXTER W. V., SCHEUERMANN T., LASTRA A.: Simulation of cloud dynamics on graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware* (2003), pp. 92–101. URL: http://dl.acm.org/citation.cfm?id=844189. 1, 2

[Hob93] HOBGOOD J. S.: Fundamentals of weather and climate. robin mcilveen, chapman and hall (london), 1992. no. of pages: xxii + 497. price: Âč19.95 (p/b). isbn 0-442-31476-0. *International Journal of Climatology 13*, 4 (1993), 461–461. URL: http://dx.doi.org/10.1002/joc.3370130409, doi:10.1002/joc.3370130409. 6

[KVH84] KAJIYA J. T., VON HERZEN B. P.: Ray tracing volume densities. *SIGGRAPH Comput. Graph. 18*, 3 (Jan. 1984), 165–174. URL: http://doi.acm.org/10.1145/964965.808594, doi:10.1145/964965.808594. 2

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), SCA, pp. 154–159. URL: http://dl.acm.org/citation.cfm?id=846276.846298. 2

[Mih11] MIHALIS L.: *First Principles of Meteorology and Air Pollution*, vol. 19. 2011. URL: http://www.springer.com/us/book/9789400701618, doi:10.1007/978-94-007-0162-5. 5

[MM13] MACKLIN M., MÜLLER M.: Position based fluids. *ACM Trans. Graph. 32*, 4 (July 2013), 104:1–104:12. URL: http://doi.acm.org/10.1145/2461912.2461984, doi:10.1145/2461912.2461984. 2

[MMPZ12] MILLER B., MUSETH K., PENNEY D., ZAFAR N. B.: Cloud Modeling And Rendering for "Puss In Boots". In *ACM SIGGRAPH Talk* (2012). URL: http://www.museth.org/Ken/Publications_files/Miller-etal_SIG12.pdf. 1

[MON05] MONAGHAN J.: Smoothed particle hydrodynamics. *Rep.Prog. Phys. 68* (2005), 1703âĂŞ1759. URL: http://iopscience.iop.org/article/10.1088/0034-4885/68/8/R01/pdf. 3

[Ney97] NEYRET F.: Qualitative simulation of cloud formation and evolution. In *8th Eurographics Workshop on Computer Animation and Simulation (EGCAS)* (sep 1997), pp. 113–124. URL: http://link.springer.com/chapter/10.1007%2F978-3-7091-6874-5_8. 2

[PH89] PERLIN K., HOFFERT E. M.: Hypertexture. *SIGGRAPH Comput. Graph. 23*, 3 (July 1989), 253–262. URL: http://doi.acm.org/10.1145/74334.74359, doi:10.1145/74334.74359. 5

[SMST04] SOARES P. M. M., MIRANDA P. M. A., SIEBESMA A. P., TEIXEIRA J.: An eddy-diffusivity/mass-flux parametrization for dry and shallow cumulus convection. *Quarterly Journal of the Royal Meteorological Society 130*, 604 (2004), 3365–3383. URL: http://dx.doi.org/10.1256/qj.03.223, doi:10.1256/qj.03.223. 3, 4, 6

[SSEH03] SCHPOK J., SIMONS J., EBERT D. S., HANSEN C.: A real-time cloud modeling, rendering, and animation system. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), SCA, pp. 160–166. URL: http://dl.acm.org/citation.cfm?id=846276.846299. 2, 3

[STW11] SCHNEIDER A. P., THOMSON T. G., WILSON M. S.: Clouds in the skies of Rio. In *ACM SIGGRAPH Talk* (2011). URL: http://webstaff.itn.liu.se/~perla/Siggraph2011/content/talks/28-schneider.pdf. 1

[WBC08] WITHER J., BOUTHORS A., CANI M.-P.: Rapid sketch modeling of clouds. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling* (Aire-la-Ville, Switzerland, Switzerland, 2008), SBM'08, Eurographics Association, pp. 113–118. URL: http://dx.doi.org/10.2312/SBM/SBM08/113-118, doi:10.2312/SBM/SBM08/113-118. 2

[Ye14] YE Z.: *Volumetric Cloud Rendering: An Animation of Clouds*. Master's thesis, Clemson University, the Netherlands, 2014. URL: http://tigerprints.clemson.edu/cgi/viewcontent.cgi?article=2926&context=all_theses. 2